Human Authentication & Authorization

Thierry Sans

Human Authentication

Intuitive definition

What is human authentication?

➡ "Determining the identity of a person"

Why would I need to authenticate you?

➡ "To be sure that you are the person that you claim to be"

Identification vs Authentication

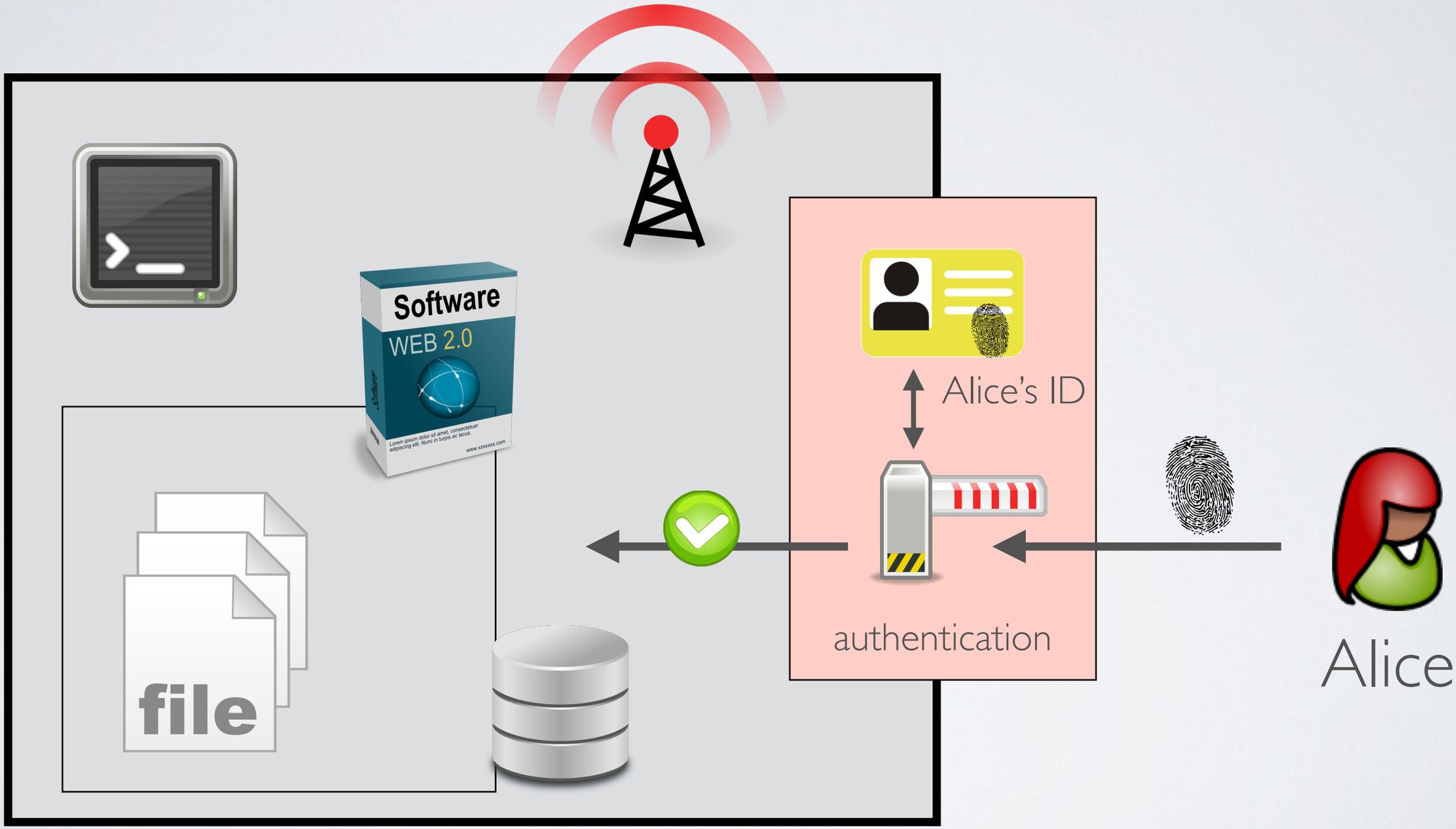
Identification

➡ Assigning a set of data to a person or an organization (subject)

Authentication

➡ Making a safe link between a subject and one or several of identities

The Big Picture



Authentication Factors

Something that you know

- ✓ Password, PIN number, secret key, secret handshake, secret questions ...

Something that you have

- ✓ IDs, badges, physical key ...

Something that you are or do (biometrics)

- ✓ Fingerprint, voice recognition, face recognition ...

Something that you know

- ✓ **Good as long as** you remember the secret and nobody can uncover or guess this secret

- ◉ **Gets compromised as soon as** someone else knows this secret and is able to use it

Something that you have

- ✓ **Good as long as** you do not lose or damage the token and there is only one instance for a "given token"

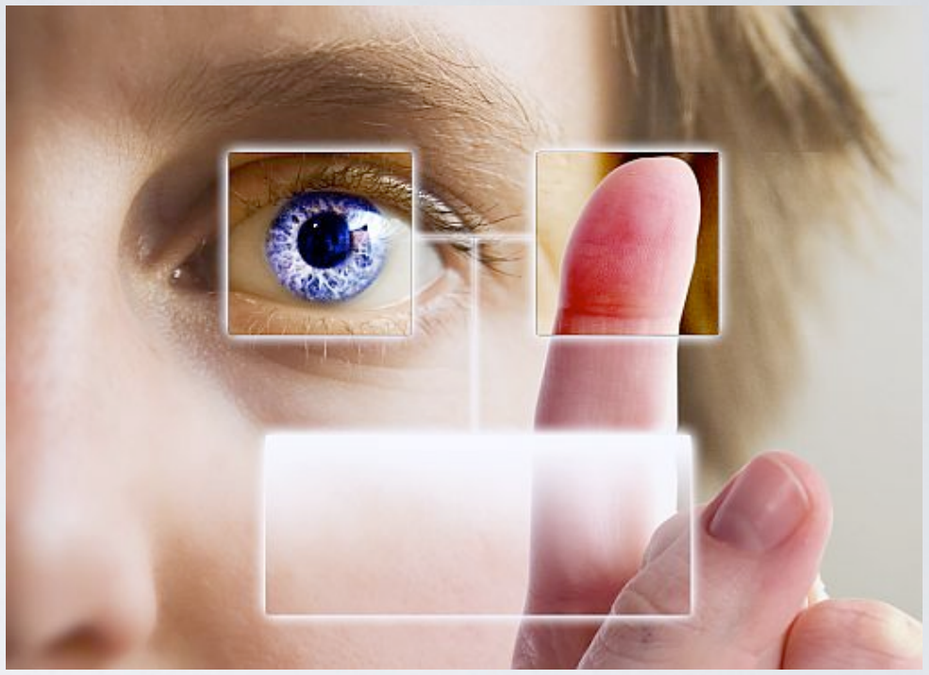
- ◉ **Gets compromised as soon as** someone can duplicate or fake the token

Something that you are or do - Biometrics

"An authenticator takes a measure of your physical characteristics and compare it with an existing measure of what you are suppose to be"

- ✓ The robustness depends on the precision of this measure and the similarity criteria (often not strict equality)
- ◉ But how to recover from an attack where the physical characteristics are compromised?

Something that you are



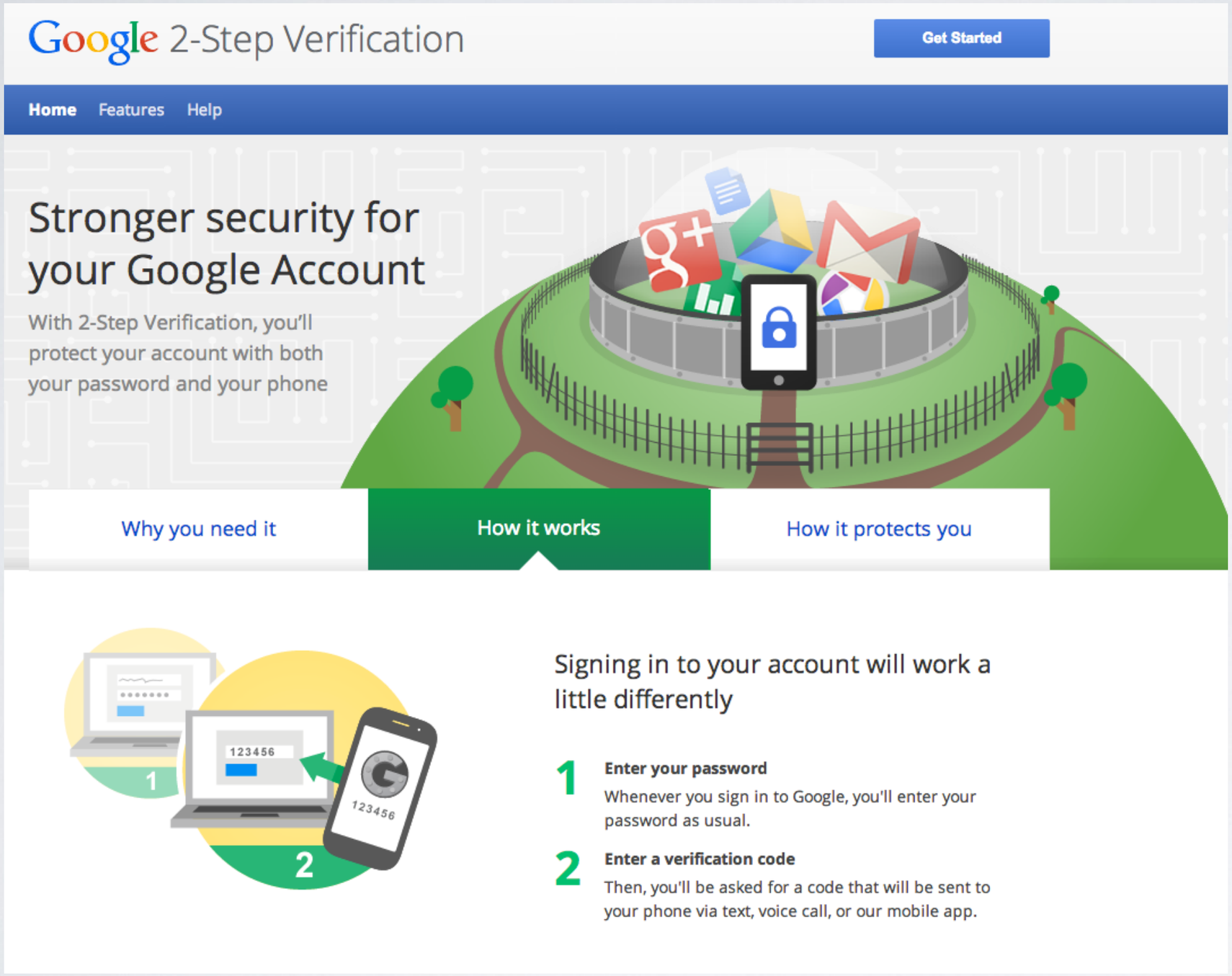
- ✓ **Good as long as** you act or look like the same and nobody cannot be "good enough" in doing what you do or "pretend" to look like you

- ◉ **Gets compromised as soon as** someone can "nearly" act like your "nearly" look like you (depending on the authenticator)

Multi-factor authentication

	Something that you		
	know	**have**	**are**
ID card		X	X
Credit Card	X	X	
Biometric Passport		X	X X
Two-factor authentication	X	X	

Example of two-factor authentication



https://twofactorauth.org/

Choosing the authentication mechanism

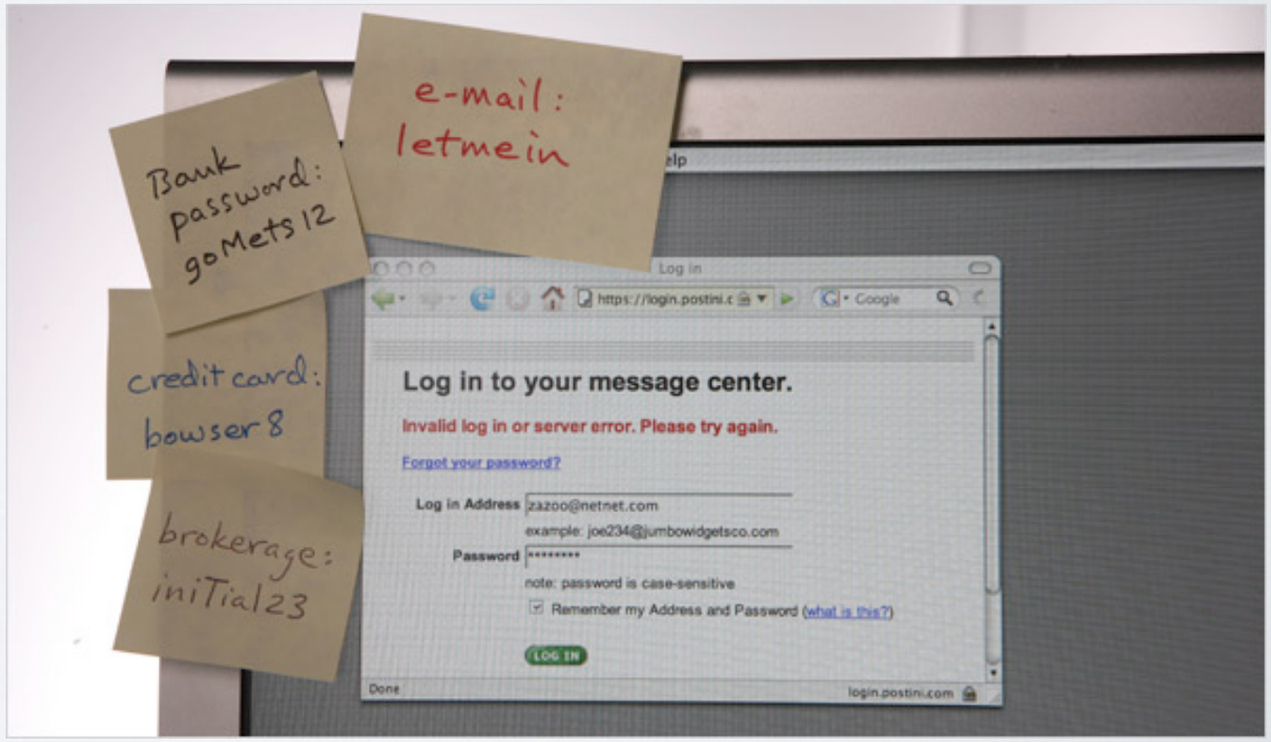
➡ Driven by the risk analysis and the costs
How hard is it to?

- Make you reveal your secret password
- Duplicate a credit card
- Fake your fingerprints

◉ There is no perfect authentication

Something else to consider - usability

- How restrictive is the use of several authentication mechanisms?
- How the users will use handle and appropriate the authentication process?



To go further

- Can the authentication process been delegate to a third part?
- Can we use the same identity over different information system?

➡ Identity management systems

Passwords

Managing Passwords

- How many passwords do you have?
- What password for what kind of application?
- How often do you change your password?
- How do you remember your password?
- How strong is your password?

Using passwords

- Where are passwords stored?
- How are they stored?
- How are they compared with an input?
- How are they transmitted on the network?

Hacking passwords

- How would you steal someone's password?
- How would you crack someone's password

Cracking a password from the login box

How to crack a password on challenge/response?

- Guessing attack (default and common passwords)
- Brute force attack
- Dictionary attack

What are the counter-measures?

- Timing
- Limit number of tries

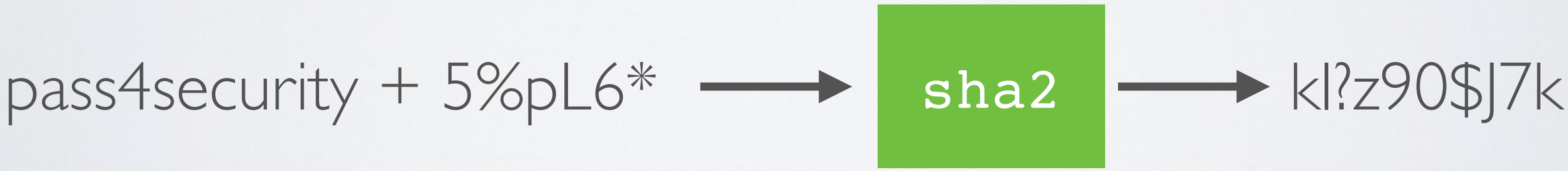
Tool : THC Hydra

How passwords are stored

- **In clear** (really bad)
- **Hashed** (bad)
- **Salted Hash** (better and easy to manage)
- **Encrypted** (best but complex to manage)

Unsalted passwords

Salted password



Getting someone's password

How to get a password in clear?

- Social engineering - Phishing
- Data mining (emails, logs)
- Keyloggers (keystroke logging)

How to get an encrypted or hashed password?

- Know where it is stored

Cracking an encrypted or hashed password

How to crack a password knowing its stored form?

- Guessing attack (default and common passwords)
- Brute force attack
- Dictionary attack
- Rainbow tables

What are the counter-measures?

- Protect it well at the OS or application level
- Store it somewhere else (portable device, kerberos, …)

Tool : John the Ripper

Password Strength

How strong is your password?

http://howsecureismypassword.net/

How long does it take to crack a password?

http://www.lockdown.co.uk/?pg=combi

Millions of LinkedIn passwords reportedly leaked online

A hacker says he's posted 6.5 million LinkedIn passwords on the Web -- hot on the heels of security researchers' warnings about privacy issues with LinkedIn's iOS app.

by Lance Whitney | June 6, 2012 6:31 AM PD

Update 1:08 p.m. PT: *LinkedIn confirms that passwords were "compromised."*

LinkedIn users could be facing yet another security problem.

A user in a Russian forum says that he has hacked and uploaded almost 6.5 million LinkedIn passwords, according to The Verge. Though his claim has yet to be confirmed, Twitter users are already reporting that they've found their hashed LinkedIn passwords on the list, security expert Per Thorsheim said.

CNET | News
Reviews | News | Download | CNET TV | How To | Deals

Hackers post 450K credentials pilfered from Yahoo

Credentials posted in plain text appear to have originated from the Web company's Yahoo Voices platform. The hackers say they intended the data dump as a "wake-up call."

by Steven Musil | July 11, 2012 11:06 PM PDT

Yahoo has been the victim of a security breach that yielded hundreds of thousands of login credentials stored in plain text.

The hacked data, posted to the hacker site D33D Company, contained more than 453,000 login credentials and appears to have originated from the Web pioneer's network. The hackers, who said they used a union-based SQL injection technique to penetrate the Yahoo subdomain, intended the data dump to be a "wake-up call."

Stronger password (used for e-banking for instance)

Visual Pad (weak)

One time password (stronger)

- Calculator
- Password sheet

Two-factor authentication (better)

- Password (something you know)
- SMS code (something you own)

Authorization
(a.k.a Access Control)

Examples

- Physical systems
- Filesystems
- Database Management System
- Web applications
- Firewall
- …

Outline

- The intuition
- The theory
- The practice

The Intuitive Approach

System, Subjects and Resources

- **The system** enables the subjects to use the resources

- **The subjects** are the active entities of the system

- **The resources** are made available by the system

Policy, Reference Monitor and Access Control Rules

- **The policy** defines who can (and sometimes how to) access the resources

- **The reference monitor** controls the access to the resources

- **The access control rules** implement the policy and are to be evaluated by the reference monitor

The room policy in the IC building

"People using the IC building are either faculty or students. Currently, there are 100 faculty and 1000 students.

The IC building has 50 rooms: 10 are accessible to faculty only, 10 are accessible to students only and 30 are accessible to both faculty and students."

rules

The Access Control Matrix

	student-lounge	*classroom*	*faculty-lounge*	...
Alice	✓	✓	✗	...
Bob	✓	✓	✗	...
Charlie	✗	✓	✓	...
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Representation of the access control matrix

The matrix can be represented as either:

- non-null triples (database style)
- access control lists (by resources)
- capability lists (by subjects)

✓ Permissions are sufficient to represent the matrix

➡ What is not explicitly allowed is denied
(closed world hypothesis)

Example of rules given as non-null triples

r1: **Alice** can open *classroom*

r2: **Alice** can open *student-lounge*

r3: **Bob** can open *classroom*

r4: **Bob** can open *student-lounge*

r5: **Charlie** can open *classroom*

r6: **Charlie** can open faculty-*lounge*

r7: **. . .**

Evaluating non-null triples

`if`

S requests to open *R*

`and`

∃ r_i | r_i: **S** can open *R*

`then`

open *R*

Example of rules as capability lists

r1: **Alice** can open *1064, student-lounge*

r2: **Bob** can open *1064, student-lounge*

r:3 **Charlie** can open *1064, meeting-room*

r4: **. . .**

Evaluating capability lists

`if`

S requests to open *R*

`and`

(∃ r_i | r_i: **S** can open by R_1 ... R_n `and` *R* ∈ *[*R_1 ... R_n*]*)

`then`

open *R*

Example of rules given as Access Control lists

r1: *1064* can be opened by **Alice, Bob, Charlie**

r2: *student-lounge* can be opened by **Alice, Bob**

r3: *meeting-room* can be opened by **Charlie**

r4: **. . .**

Evaluating access control lists

`if`

S requests to open *R*

`and`

(∃ r_i | r_i: *R* can be opened by S_1 ... S_n `and` **S** ∈ [S_1 ... S_n])

`then`

open *R*

The concept of role

- The permission to access to resources is mediated by a role

  S in role R has all the privilege P

Example of role-based rules

ra1: **Alice** has role student

ra2: **Bob** has role student

ra3: **Charlie** has role faculty

ra4: . . .

p1: student *can open 1064*

p2: student *can open student-lounge*

p4: faculty *can open 1064*

p5: faculty *can open meeting room*

p6: . . .

Evaluating role-based rules

`if`

S requests to open *R*

`and`

(∃ ro, ra_i and p_j | ra_i: **S** has role ro `and` p_j: ro can open *R*)

`then`

open *R*

The cost of managing the policy

For each model,

- how many rules are needed to enforce the policy?
- what are the consequences when:
  - 1 room is closed for maintenance?
  - 100 students graduate?
  - 100 new students are enrolled?
  - 1 new classroom is created?
  - 1 new lab room is created for students and faculty that are doing research?

What do we observe?

- ✓ All models implements the same policy represented by the Access Control Matrix
- ✓ Role-Based model has less rules, easier to manage

More advanced models

Constraints

The classrooms can be access **between 8am and 8pm**

The video can be accessed **only if the person is in Canada**

Separation of duties - conflict of interest

In court, the defense lawyer and the prosecution lawyer **cannot access the same pieces of information**

History based

A rented movie can be played **only once**

Self-declared constraints

A nurse can have access to the patient medical record **in case of emergency**

Administration

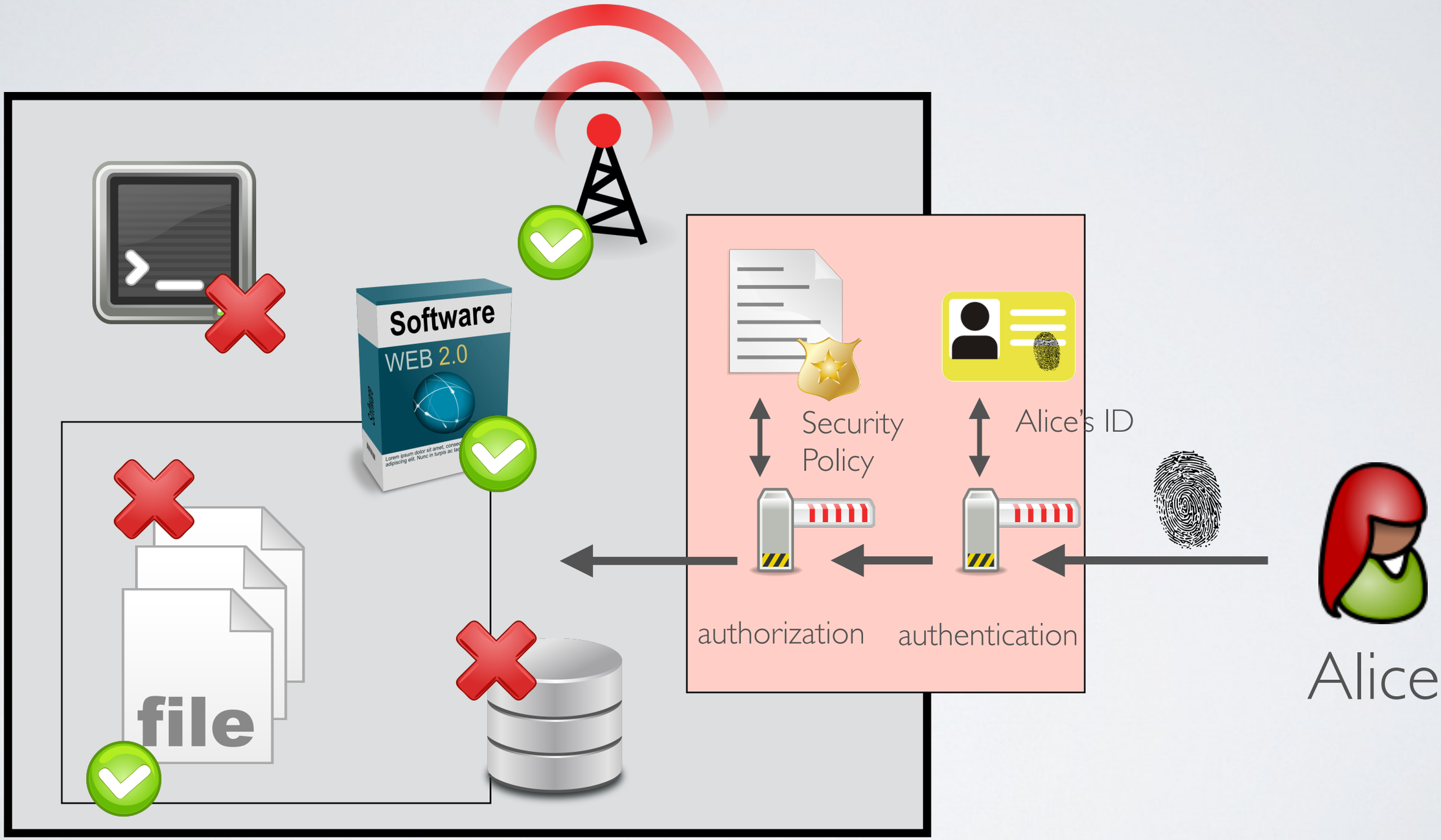
- Who can **create** a resource in the system?
- Who can **assign** and **revoke** the rights?
- Is it possible to **transfer** or **delegate** a right to someone else?

What do we observe?

- ✓ There is not one access control model but many depending on the application and the policy

Theory

The Big Picture



Access Control in the Literature

- **Subject** is the active entity of the information system
- **Object (or resource)** is a source of information managed by the information system
- **Action (or right)** produces a result which might disclose or modify the object and/or modify the information system state

was implicit in the intuitive approach

Classic Example - A Filesystem

Subjects	username
Objects	files
Actions	read, write, execute, delete, copy, move, create ...

Governing Principles

Complete mediation

➡ Every access to every object must be mediated

Least privilege

➡ Do not grant subjects more rights than they need

Specification, Implementation and Validation

	Security
Specification	Risk Analysis & Security Policy
Implementation	Access Control Mechanisms
Validation	Accounting & Audit

- Who are the users?
- What are the resources?
- What are the operations?
- What is the policy?

Choose the adequate mechanism to enforce the policy?

Does the access control mechanism reflect the security policy?

Access Control Matrix

1971 - Butler Lampson

	Domain 1	Domain 2	Domain 3	File 1	File 2	Process 1
Domain 1	*owner control	*owner control	*call	*owner *read *write		
Domain 2			call	*read	write	wakeup
Domain 3			owner control	read	*owner	

Extensions

Graham-Denning (1972)
and **Harison-Ruzzo-Ullman HRU** (1976)

➡ Creation and deletion of objects

Take-Grant model (1977 - Lipton and Snyder)

➡ Formalization of the ownership principle

Discretionary Access Control Model (DAC)

1985 - *Trusted Computer System Evaluation* - DOD

Core model

➡ Access Control Matrix

Administration model

➡ based on the ownership principle

Role-based Access Control Model (RBAC)

1992 - Ferraiolo and Kuhn

➡ Concepts of role and role hierarchy

➡ Powerful administration model called ARBAC

✓ Lower the number of rules and simplifies administration

➡ Concept of sessions

✓ Separation of privileges

Attacks

Incomplete Mediation

A **misconfiguration** in the system allows an attacker to do something that the abstract policy does not allow

Privilege Escalation

A **vulnerability** in the system allows an attacker to gain privileges that the abstract policy does not allow

Access Control in Practice